

WHAT IS CLAIMED IS:

1 1. A computer implemented method for grouping processors
2 in a computer environment that includes a plurality of
3 dissimilar processors, said method comprising:
4 receiving a request from an application that is
5 running on a first processor type;
6 assigning one or more second processor types and a
7 memory space to a group in response to the request,
8 wherein the first processor type and the assigned
9 second processor types are heterogeneous; and
10 processing an application execution thread using the
11 group, the application execution thread running on the
12 first processor type and corresponding to the
13 application.

1 2. The method as described in claim 1 wherein the first
2 processor type shares the memory space with the
3 assigned second processor types.

1 3. The method as described in claim 1 further comprising:
2 identifying whether the application requests the
3 memory space to be a private memory, wherein the
4 private memory is accessible only by the assigned
5 second processor types; and
6 classifying the memory space as the private memory.

1 4. The method as described in claim 3 further comprising:
2 retrieving data from the private memory using one of
3 the assigned second processor types;

4 manipulating the data using one of the assigned second
5 processor types, the manipulating resulting in
6 resultant data; and

7 storing the resultant data in a shared memory, the
8 shared memory accessible by the first processor type.

1 5. The method as described in claim 1 further comprising:
2 retrieving an affinity selection bit from the
3 application;

4 determining whether the application requests affinity
5 processor selection based upon the affinity selection
6 bit; and

7 performing the assigning using affinity processor
8 selection.

1 6. The method as described in claim 5 wherein the
2 performing further comprises:

3 selecting one of the second processor types based upon
4 the affinity processor selection;

5 determining whether the selected second processor type
6 is available; and

7 performing the assigning based upon the selected
8 second processor type's availability.

1 7. The method as described in claim 1 further comprising:
2 detecting that one or more of the second processor
3 types are in use by an active execution thread;

4 identifying an active priority that corresponds to the
5 active execution thread;

6 comparing the active priority to a requesting
7 priority, the requesting priority corresponding to the
8 application execution thread; and

9 terminating the active execution thread if the active
10 priority is lower than the requesting priority.

1 8. The method as described in claim 1 wherein the group
2 corresponds to one or more group properties, wherein
3 the group properties are selected from the group
4 consisting of a sharing mode, a priority, and a
5 scheduling policy.

1 9. The method as described in claim 1 wherein the group
2 includes a plurality of second processors.

1 10. The method as described in claim 1 wherein the first
2 processor type is a processing unit and wherein the
3 second processor types are synergistic processing
4 units.

1 11. An information handling system comprising:
2 a plurality of dissimilar processors;
3 a memory accessible by the plurality of dissimilar
4 processors;
5 one or more nonvolatile storage devices accessible by
6 the plurality of dissimilar processors; and
7 a processor grouping tool for compiling source code,
8 the processor grouping tool comprising software code
9 effective to:

10 receive a request from an application that
11 is running on a first processor type, the

12 first processor type included in the
13 plurality of dissimilar processors;
14 assign one or more second processor types
15 included in the plurality of dissimilar
16 processor types and a memory space included
17 in the memory to a group in response to the
18 request, wherein the first processor type
19 and the assigned second processor types are
20 heterogeneous; and
21 process an application execution thread
22 using the group, the application execution
23 thread running on the first processor type
24 and corresponding to the application.

1 12. The information handling system as described in claim
2 11 wherein the first processor type shares the memory
3 space with the assigned second processor types.

1 13. The information handling system as described in claim
2 11 wherein the software code is further effective to:
3 identify whether the application requests the memory
4 space to be a private memory, wherein the private
5 memory is accessible only by the assigned second
6 processor types; and
7 classify the memory space as the private memory.

1 14. The information handling system as described in claim
2 13 wherein the software code is further effective to:
3 retrieve data from the private memory using one of the
4 assigned second processor types;

5 manipulate the data using one of the assigned second
6 processor types whereby the data manipulation results
7 in resultant data; and

8 store the resultant data in a shared memory included
9 in the memory, the shared memory accessible by the
10 first processor type.

1 15. The information handling system as described in claim
2 11 wherein the software code is further effective to:
3 retrieve an affinity selection bit from the
4 application;

5 determine whether the application requests affinity
6 processor selection based upon the affinity selection
7 bit; and

8 perform the assigning using affinity processor
9 selection.

1 16. The information handling system as described in claim
2 15 wherein the software code is further effective to:
3 select one of the second processor types based upon
4 the affinity processor selection;

5 determine whether the selected second processor type
6 is available; and

7 perform the assigning based upon the selected second
8 processor type's availability.

1 17. The information handling system as described in claim
2 11 wherein the software code is further effective to:
3 detect that one or more of the second processor types
4 are in use by an active execution thread;

5 identify an active priority that corresponds to the
6 active execution thread;

7 compare the active priority to a requesting priority,
8 the requesting priority corresponding to the
9 application execution thread; and

10 terminate the active execution thread if the active
11 priority is lower than the requesting priority.

1 18. The information handling system as described in claim
2 11 wherein the group corresponds to one or more group
3 properties, wherein the group properties are selected
4 from the group consisting of a sharing mode, a
5 priority, and a scheduling policy.

1 19. The information handling system as described in claim
2 11 wherein the group includes a plurality of second
3 processors.

1 20. The information handling system as described in claim
2 11 wherein the first processor type is a processing
3 unit and wherein the second processor types are
4 synergistic processing units.

1 21. A computer program product stored on a computer
2 operable media for grouping processors for a plurality
3 of dissimilar processors, said computer program
4 product comprising:

5 means for receiving a request from an application that
6 is running on a first processor type;

7 means for assigning one or more second processor types
8 and a memory space to a group in response to the

9 request, wherein the first processor type and the
10 assigned second processor types are heterogeneous; and
11 means for processing an application execution thread
12 using the group, the application execution thread
13 running on the first processor type and corresponding
14 to the application.

1 22. The computer program product as described in claim 21
2 wherein the first processor type shares the memory
3 space with the assigned second processor types.

1 23. The computer program product as described in claim 21
2 further comprising:
3 means for identifying whether the application requests
4 the memory space to be a private memory, wherein the
5 private memory is accessible only by the assigned
6 second processor types; and

7 means for classifying the memory space as the private
8 memory.

1 24. The computer program product as described in claim 23
2 further comprising:

3 means for retrieving data from the private memory
4 using one of the assigned second processor types;

5 means for manipulating the data using one of the
6 assigned second processor types, the manipulating
7 resulting in resultant data; and

8 means for storing the resultant data in a shared
9 memory, the shared memory accessible by the first
10 processor type.

1 25. The computer program product as described in claim 21
2 further comprising:
3 means for retrieving an affinity selection bit from
4 the application;
5 means for determining whether the application requests
6 affinity processor selection based upon the affinity
7 selection bit; and
8 means for performing the assigning using affinity
9 processor selection.

1 26. The computer program product as described in claim 25
2 wherein the performing further comprises:
3 means for selecting one of the second processor types
4 based upon the affinity processor selection;
5 means for determining whether the selected second
6 processor type is available; and
7 means for performing the assigning based upon the
8 selected second processor type's availability.

1 27. The computer program product as described in claim 21
2 further comprising:
3 means for detecting that one or more of the second
4 processor types are in use by an active execution
5 thread;
6 means for identifying an active priority that
7 corresponds to the active execution thread;
8 means for comparing the active priority to a
9 requesting priority, the requesting priority
10 corresponding to the application execution thread; and

11 means for terminating the active execution thread if
12 the active priority is lower than the requesting
13 priority.

1 28. The computer program product as described in claim 21
2 wherein the group corresponds to one or more group
3 properties, wherein the group properties are selected
4 from the group consisting of a sharing mode, a
5 priority, and a scheduling policy.

1 29. The computer program product as described in claim 21
2 wherein the group includes a plurality of second
3 processors.

1 30. The computer program product as described in claim 21
2 wherein the first processor type is a processing unit
3 and wherein the second processor types are synergistic
4 processing units.
